

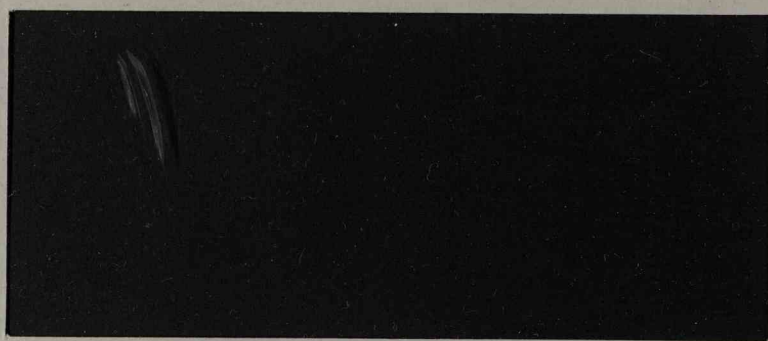


WORKING PAPERS

UN MODELLO URBANO A LARGA SCALA PER L'AREA METROPOLITANA DI TORINO

C. S. Bertuglia, S. Occelli, G. A. Rabino
R. Tadei, C. Salomone

WP n. 1



INDICE

Working Papers n. 1

1. INTRODUZIONE

2. STRUTTURA TEORICA E FORMULAZIONE OPERATIVA DEL MODELLO

2.1. STRUTTURA TEORICA DEL MODELLO

2.1.1. Introduzione

UN MODELLO URBANO A LARGA SCALA PER L'AREA METROPOLITANA DI TORINO

C. S. Bertuglia, S. Occelli, G. A. Rabino
R. Tadei, C. Salomone

2.2. FORMULAZIONE OPERATIVA DEL MODELLO

WP n. 1

2.2.1. Problemi di costruzione del software di un modello a grandi dimensioni

2.2.2. L'organizzazione delle informazioni nei modelli a grandi dimensioni

BIBLIOGRAFIA

Gennaio 1981

Lavoro presentato alla Conferenza Italiana di
Scienze Regionali, Roma, 24-26/12/1980

I N D I C E

1. INTRODUZIONE
2. STRUTTURA TEORICA E FORMULAZIONE OPERATIVA DEL MODELLO
 - 2.1. STRUTTURA TEORICA DEL MODELLO
 - 2.1.1. Introduzione
 - 2.1.2. Il modello di simulazione
 - 2.1.3. Approccio fuzzy alla funzione di utilità
 - 2.1.4. Stabilità strutturale
 - 2.2. FORMULAZIONE OPERATIVA DEL MODELLO
 - 2.2.1. Problemi di costruzione del software di un modello a grandi dimensioni
 - 2.2.2. L'organizzazione delle informazioni nei modelli a grandi dimensioni

BIBLIOGRAFIA

1. INTRODUZIONE

In questa comunicazione si presenta un modello di simulazione dell'evoluzione del sistema urbano di Torino, messo a punto per sondare gli effetti delle politiche di intervento settoriale (industria, terziario, trasporti, uso del suolo ecc.), che dovrebbero concorrere a definire il nuovo Piano Regolatore Generale della Città di Torino.

La prima parte della comunicazione è dedicata all'illustrazione della struttura teorica del modello. Data la complessità dello stesso, viene descritto in dettaglio solo il nucleo centrale, del modello, e precisamente il sottomodello di localizzazione residenziale. Per altro, tale sottomodello è anche quello che offre maggiore interesse, sia per gli elementi di novità presenti (approccio fuzzy alla funzione di utilità di localizzazione), sia per l'analisi della stabilità delle soluzioni che si può compiere su di esso.

La seconda parte della comunicazione è poi dedicata ai problemi dell'implementazione operativa del modello. Sono illustrate le soluzioni adottate sia per la programmazione all'elaboratore elettronico di un modello a larga scala sia per la gestione degli input dello stesso.

La comunicazione descrive poi i problemi e le soluzioni adottate per la costruzione degli input necessari al modello.

2. STRUTTURA TEORICA E FORMULAZIONE OPERATIVA DEL MODELLO

Il modello, di cui in questa comunicazione, è in corso di sperimentazione.

2.1.1. Introduzione

La struttura teorica del modello è fondata, per gli aspetti socio-economici, sulle schemi concettuali del modello di Lowry (Lowry, 1954) e, per gli aspetti territoriali, sul metodo della massimizzazione dell'entropia di Wilson (Wilson, 1967). La formulazione matematica si ispira poi, per gli aspetti dinamici, al modello di Forrester (Forrester, 1969).

I sottosistemi presi in considerazione nel modello sono, cioè, i sottosistemi di cui il sistema è composto e che sono oggetto di studio sono (cfr. Fig. 1):

1. il sottosistema industriale;
2. il sottosistema terziario;
3. il sottosistema delle popolazioni;
4. il sottosistema delle abitazioni;
5. il sottosistema residenziale (che deriva, come noto, dalla interazione fra i due sottosistemi di cui sub 1. e sub 4.);
6. il sottosistema dei trasporti;
7. il sottosistema di uso del suolo (a).

(a) Per una descrizione più ampia ed articolata del modello, si rimanda a Bertuglio, Cecchi, Marini, Tassi (1980).

2. STRUTTURA TEORICA E FORMULAZIONE OPERATIVA DEL MODELLO

2.1. STRUTTURA TEORICA DEL MODELLO

2.1.1. Introduzione

La struttura teorica del modello è fondata, per gli aspetti socioeconomici, sullo schema causale del modello di Lowry (Lowry, 1964) e, per gli aspetti territoriali, sul metodo della massimizzazione dell'entropia di Wilson (Wilson, 1970); la formalizzazione matematica si ispira poi, per gli aspetti dinamici, al modello di Forrester (Forrester, 1969).

I sottosistemi presi in considerazione nel modello -e cioè, i sottosistemi nei quali è stato articolato il sistema oggetto di studio- sono (cfr.: fig. 1):

1. il sottosistema industriale;
2. il sottosistema terziario;
3. il sottosistema della popolazione;
4. il sottosistema delle abitazioni;
5. il sottosistema residenziale (che deriva, come noto, dalla interrelazione fra i due sottosistemi di cui sub 3. e sub 4.);
6. il sottosistema dei trasporti;
7. il sottosistema di uso del suolo (*).

(*) Per una descrizione più ampia ed articolata dei singoli sottosistemi, si rimanda a: Bertuglia, Occeili, Rabino, Tadei (1980).

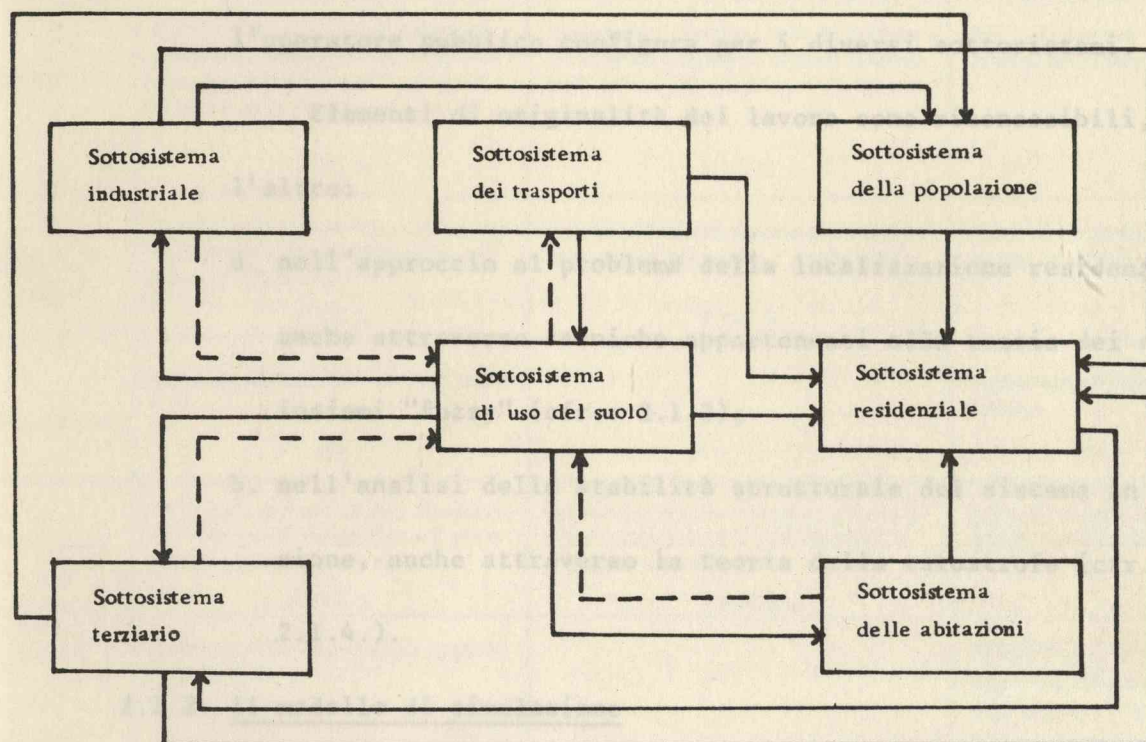


Figura 1 - Sottosistemi del sistema urbano e loro principali interazioni

- Principali interazioni dirette fra i sottosistemi
- - - - Principali interazioni di feedback fra i sottosistemi

Di detti sottosistemi il modello cerca di descrivere l'evoluzione, simulando sia i principali fenomeni responsabili della crescita o della stagnazione sia l'impatto delle politiche che l'operatore pubblico configura per i diversi sottosistemi.

Elementi di originalità del lavoro sono riconoscibili, tra l'altro:

- a. nell'approccio al problema della localizzazione residenziale, anche attraverso tecniche appartenenti alla teoria dei sottoinsiemi "fuzzy" (cfr.: 2.1.3);
- b. nell'analisi della stabilità strutturale del sistema in discussione, anche attraverso la teoria della catastrofe (cfr.: 2.1.4.).

2.1.2. Il modello di simulazione

Nella tabella di fig. 2, per ciascuno dei sottosistemi considerati -cui è associato il relativo sottomodello- (cfr.: fig.1) vengono riassunti i principali fenomeni e le principali politiche simulate dal modello; si riportano, inoltre, le principali variabili descrittive dei sottosistemi stessi.

Sottosistema	Fenomeni simulati dal modello	Politiche simulate dal modello	Variabili descrittive del sottosistema (variabili di livello)
1. industriale	apertura, chiusura, rilocalizzazione di stabilimenti industriali e variazione del numero di posti di lavoro negli stabilimenti esistenti, per settore t e zona j $F I(j, t)$	costruzione e rilocalizzazione di stabilimenti industriali e variazioni di posti di lavoro negli stabilimenti, per settore t e zona j $P I(j, t)$	IND (j, t): numero di posti di lavoro industriali nel settore t e nella zona j
2. terziario	rilocalizzazione di servizi terziari e fenomeni diversi dalla rilocalizzazione, nel settore l e nella zona j $F T(j, l)$	variazione e rilocalizzazione di posti di lavoro terziari, nel settore l e nella zona j $P T(j, l)$	TERZ (j, l): numero di posti di lavoro terziari nel settore l e nella zona j
3. della popolazione	variazione demografica naturale e migratoria $F D$		P: popolazione totale per il complesso dell'area
4. delle abitazioni	costruzione di nuovi alloggi, demolizione e recupero di alloggi obsoleti, per tipologia residenziale s e zona i $F A(i, s)$	costruzione di nuovi alloggi ed intervento sul patrimonio esistente per tipologia residenziale s e zona i $P A(i, s)$	AB (i, s): numero di alloggi secondo la tipologia residenziale s , nella zona i
5. residenziale	variazione nella distribuzione spaziale delle famiglie		DPOTO (i, j, f, s): numero di famiglie di tipo f , con capofamiglia occupato nella zona di lavoro j , che risiedono in alloggi di tipo s , nella zona residenziale i . DPOTD (i, g, s): numero di famiglie di tipo g , con capofamiglia non occupato, che risiedono in alloggi di tipo s , nella zona residenziale i
6. dei trasporti	variazione dell'accessibilità delle zone	politiche di trasporto per i viaggi dalla zona i alla zona j , secondo il mezzo di trasporto v $P T R(i, j, v)$	T (i, j, v): tempo di viaggio tra la zona i e la zona j , secondo il mezzo di trasporto v
7. dell'uso del suolo	variazione dell'uso del suolo nella zona j (variazione dell'attrattività delle zone) $F S(j)$	programmi di uso del suolo industriale, residenziale, terziario e residuo (verde, strade ecc.), nella zona j $P S(j)$	SLI (j): mq di suolo industriale nella zona j SLA (j): mq di suolo residenziale nella zona j SLT (j): mq di suolo terziario nella zona j R (j): mq di suolo residuo nella zona j

Figura 2 - Tabella riassuntiva dei fenomeni e delle politiche simulate dal modello ed elenco delle principali variabili del modello

Il funzionamento del modello complessivo può essere sintetizzato nelle sue linee essenziali dallo schema di fig. 3 (*).

Come accennato, il modello cerca di descrivere l'evoluzione del sistema urbano torinese (studiando appunto l'andamento, nel tempo, delle variabili di livello del modello), simulando i fenomeni responsabili della crescita o della stagnazione dei diversi sottosistemi e le politiche configurate dall'operatore pubblico, descritti nella tabella di fig. 2.

Come si può trarre dallo schema di fig. 3, la struttura del modello nel suo complesso è relativamente semplice. Eccetto i sottomodelli di uso del suolo e di trasporto (nonché di localizzazione residenziale, del quale si tratterà ampiamente più avanti), il cui funzionamento è, di per se stesso, maggiormente interconnesso a quello degli altri, gli altri sottomodelli (industriale, terziario, della popolazione e delle abitazioni) presentano una relativa autonomia. Esistono, inoltre, poche e fondamentali interazioni fra i sottomodelli stessi (cfr.: fig.3). In particolare, fra queste interazioni è possibile riconoscere la relazione causale del modello di Lowry ($IND \rightarrow P \rightarrow DPOTO$, $DPOTD \rightarrow TERZ \rightarrow P$), le intera-

(*) Per uno schema completo del modello e delle sue equazioni, si rimanda a: Bertuglia, Occelli, Rabino, Tadei (1980).

dei simboli utilizzati, si veda la fig. 2)

----- Principali interazioni dirette

---- Principali interazioni di feedback

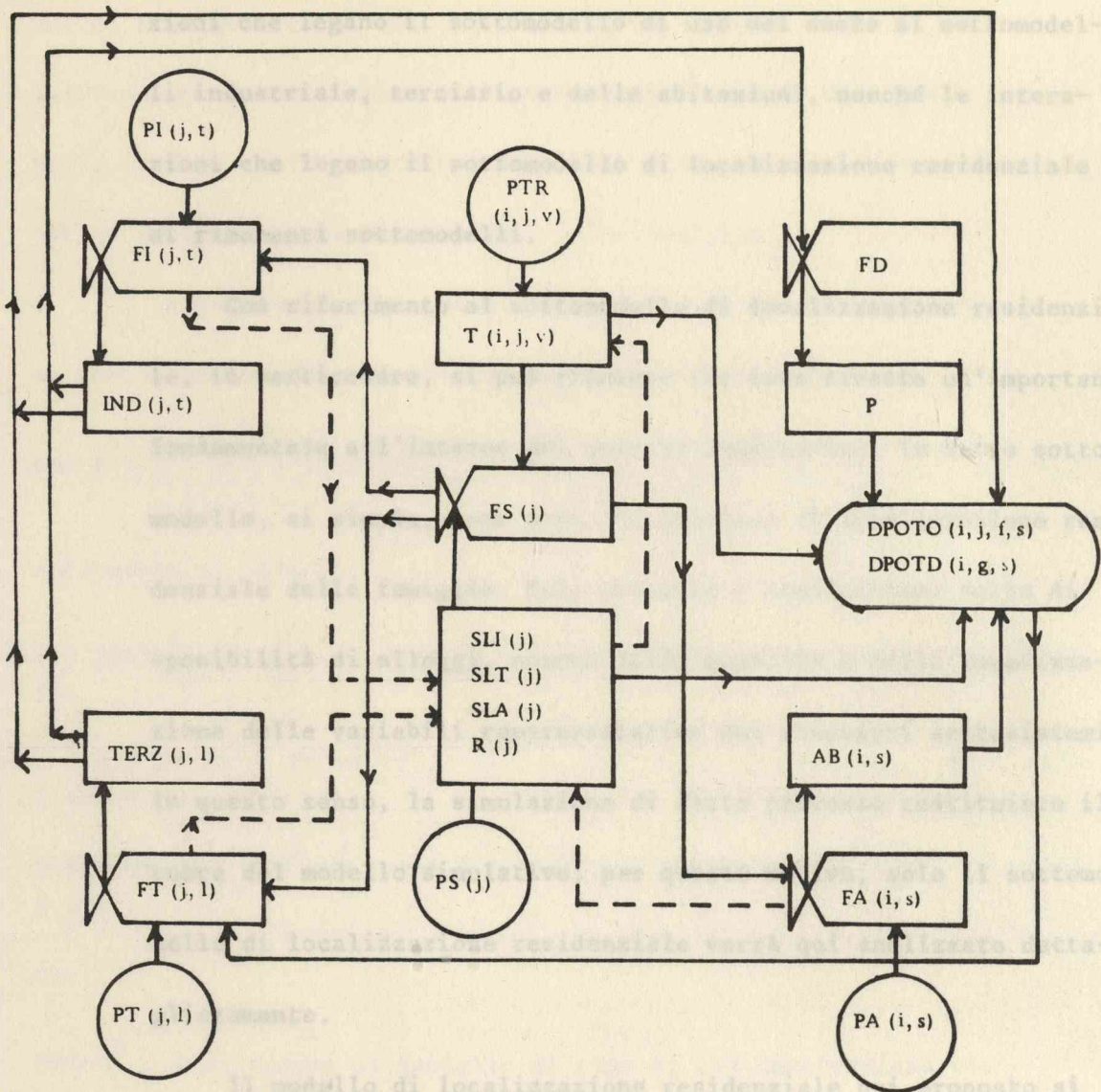


Figura 3 - Schema del modello complessivo (per la spiegazione dei simboli utilizzati, si veda la fig. 2)

- Principali interazioni dirette
- Principali interazioni di feedback

zioni che legano il sottomodello di uso del suolo ai sottomodelli industriale, terziario e delle abitazioni, nonché le interazioni che legano il sottomodello di localizzazione residenziale ai rimanenti sottomodelli.

Con riferimento al sottomodello di localizzazione residenziale, in particolare, si può rilevare che esso riveste un'importanza fondamentale all'interno del modello complessivo. In detto sottomodello, si simula, come noto, il processo di localizzazione residenziale delle famiglie. Tale processo è condizionato dalla disponibilità di alloggi, nonché dalla quantità e dalla localizzazione delle variabili rappresentative dei rimanenti sottosistemi. In questo senso, la simulazione di detto processo costituisce il cuore del modello simulativo; per questo motivo, solo il sottomodello di localizzazione residenziale verrà qui analizzato dettagliatamente.

Il modello di localizzazione residenziale qui proposto si ispira al modello di Anas (Anas, 1973). In detto modello, viene a cadere l'ipotesi della concorrenza perfetta del mercato, ipotesi presente nei modelli di Alonso (Alonso, 1964) e di Herbert e Stevens (Herbert e Stevens, 1960). In altri termini, non tutte le famiglie trovano la localizzazione residenziale ottimale: alcune pero vengono ad un livello di utilità maggiore del livello atteso, al-

tre ne rimangono al di sotto. Il modello in oggetto tenta di descrivere un mercato reale, utilizzando una misura di disequilibrio, fondata sullo scarto tra l'utilità attesa e l'utilità reale per la localizzazione residenziale delle famiglie.

Il sottomodello, che qui si introduce, ricerca la localizzazione residenziale più probabile massimizzando l'entropia S (*):

$$\max S = - \sum_i \sum_j \sum_f \sum_s \ln DPOTO_{ijs}^f !, \quad (1)$$

sottoposta ai vincoli:

$$\sum_i \sum_s DPOTO_{ijs}^f = Q_j^f \quad j=1,2,\dots,J; \quad f=1,2,\dots,F \quad (2)$$

$$\sum_i \sum_j \sum_s \sum_v DPOTO_{ijs}^f TPROB_v^f t_{ij}^v = T^f \quad f=1,2,\dots,F \quad v=1,2,\dots,V \quad (3)$$

$$\sum_i \sum_j \sum_s DPOTO_{ijs}^f (\bar{U}^f - U_{is}^f) = \Delta U^f \quad f=1,2,\dots,F, \quad (4)$$

ove:

$DPOTO_{ijs}^f$ è il numero di famiglie di tipo f , con capofamiglia occupato, che risiedono in alloggi di tipo s nella zona i e con posto di lavoro nella zona j ;

Q_j^f è il numero di famiglie di tipo f con posto di lavoro nella zona j ;

t_{ij}^v è il costo di trasporto tra la zona i e la zona j , se-

(*) Solo le famiglie con capofamiglia occupato vengono localizzate con il metodo della massimizzazione dell'entropia.

condo il mezzo di trasporto v;

T^f è il costo totale dei viaggi per l'insieme delle famiglie di tipo f;

$TPROB_v^f$ è la probabilità che una famiglia di tipo f utilizzi il mezzo di trasporto v;

\bar{U}^f è il valore di utilità atteso per una famiglia di tipo f;

U_{is}^f è il valore di utilità reale per una famiglia di tipo f che sceglie un alloggio di tipo s, nella zona residenziale i.

La soluzione del sistema (1), (2), (3) e (4) si ottiene co-

struendo il lagrangiano :

$$L = - \sum_i \sum_j \sum_f \sum_s \ln DPOTO_{ijs}^f + \\ - \sum_j \sum_f \lambda_j^f \left(\sum_i \sum_s DPOTO_{ijs}^f - Q_j^f \right) + \\ - \sum_f \beta^f \left(\sum_i \sum_j \sum_s DPOTO_{ijs}^f TPROB_v^f t_{ij}^v - T^f \right) + \\ - \sum_f \xi^f \left[\sum_i \sum_j \sum_s DPOTO_{ijs}^f (\bar{U}^f - U_{is}^f) - \Delta U^f \right],$$

ove λ_j^f , β^f , ξ^f sono i moltiplicatori di Lagrange associati, rispettivamente, ai vincoli (2), (3) e (4),

e risolvendo

$$\frac{\partial L}{\partial DPOTO_{ijs}^f} = 0.$$

Si ottiene:

$$DPOTO_{ijs}^f = \sum_v B_j^f Q_j^f TPROB_v^f \cdot \exp(-\beta^f t_{ij}^v) \exp[-\xi^f (\bar{U}^f - U_{is}^f)], \quad (5)$$

ove

$$B_j^f = \frac{\exp(-\lambda_j^f)}{Q_j^f}.$$

Usando l'equazione (2) si ottiene la formulazione esplicita

di B_j^f :

$$B_j^f = \left\{ \sum_i \sum_s \sum_v TPROB_v^f \exp(-\beta^f \cdot t_{ij}^v) \exp[-\xi^f (\bar{U}^f - U_{is}^f)] \right\}^{-1}.$$

Una peculiarità del sottomodello di localizzazione residenziale consiste nel simulare, per ogni periodo di simulazione, la mobilità residenziale di tutte le famiglie con capofamiglia occupato. Ciò permette di individuare le famiglie che cambiano residenza, in modo più preciso di quanto avverrebbe localizzando solo le famiglie che subiscono variazioni occupazionali (*); infatti, quest'ultimo approccio richiederebbe in più l'introduzione di

(*) Le famiglie che non subiscono variazioni occupazionali vengono riassegnate alla stessa localizzazione residenziale del periodo di simulazione precedente.

un modello di mobilità residenziale.

Rispetto alla formulazione di Anas, questo sottomodello di localizzazione residenziale presenta due elementi di novità:

a. i tempi di viaggio (o, se si vuole, i costi di trasporto), per recarsi dall'origine i alla destinazione j , non compaiono nella funzione di utilità delle famiglie e devono sottostare al vincolo relativo al tempo di viaggio totale;

b. la funzione di utilità (reale) delle famiglie viene definita mediante la teoria dei sottoinsiemi fuzzy. Si ritiene, infatti, che detta funzione risenta di una certa imprecisione [dovuta, ad esempio, alla difficoltà da parte delle famiglie di scegliere tra differenti beni (Ponsard, 1978)] ed è configurata come funzione di più variabili, valutate secondo criteri soggettivi.

Questo argomento sarà approfondito in 2.1.3..

2.1.3. Approccio fuzzy alla funzione di utilità

Come introdotto in 2.1.2., l'utilità familiare -e, cioè, il soddisfacimento che una famiglia raggiungerebbe qualora scegliesse una certa localizzazione residenziale- non è, nella realtà, oggettivamente rilevabile e quantificabile. In primo luogo, perché le famiglie non possiedono una conoscenza perfetta delle

caratteristiche delle diverse localizzazioni residenziali (e queste, dunque, costituiscono, dal punto di vista delle famiglie, uno spazio economico impreciso); in secondo luogo, perché l'utilità è, di per se stessa, una nozione multidimensionale e soggettiva; infatti, la scelta localizzativa da parte delle famiglie è motivata da un insieme di fattori, per ciascuno dei quali le famiglie esprimono una valutazione soggettiva (Fustier, 1978).

Posto quanto sopra, si può supporre che, per una famiglia di tipo f , ogni localizzazione residenziale sia caratterizzata dal seguente insieme di fattori localizzativi:

M_{is}^f = quantità di denaro, destinata a spese non residenziali, da parte di una famiglia di tipo f che occupa un alloggio

di tipo s nella zona i ;

H_{is} = numero di alloggi di tipo s nella zona i ;

A_i^f = accessibilità relativa della zona i per una famiglia di tipo f ;

$N_{i(\ell)}$ = insieme delle caratteristiche ambientali ($\ell=1, \dots, L$) della zona i .

L'utilità procurata ad una famiglia di tipo f , dalla scelta di una localizzazione residenziale, può essere espressa come:

$$U_{is}^f = g \left[\bar{M}_{is}^f, H_{is}^f, A_i^f, N_{i(\ell)}^f; \gamma_{is}^M(f), \gamma_{is}^H(f), \gamma_i^A(f), \gamma_{i(\ell)}^N(f) \right], \quad (6)$$

ove:

U_{is}^f è l'utilità procurata ad una famiglia di tipo f dalla scelta della zona residenziale i e della tipologia residenziale s ;
 g è la relazione funzionale;

$\bar{M}_{is}^f, H_{is}^f, A_i^f, N_{i(\ell)}^f$ sono i fattori localizzativi definiti più sopra;
 $\gamma_{is}^M(f), \gamma_{is}^H(f), \gamma_i^A(f), \gamma_{i(\ell)}^N(f)$ sono i pesi che una famiglia di tipo f esprime per ciascuno dei fattori localizzativi prima elencati.

La funzione di utilità (6) può essere chiamata funzione di utilità fuzzy.

Esplicitando la (6), la funzione di utilità fuzzy può essere riscritta come segue:

$$U_{is}^f = \gamma_{is}^M(f) \cdot m_{is}^f + \gamma_{is}^H(f) \cdot h_{is} + \gamma_i^A(f) \cdot a_i^f + \sum_{\ell=1}^L \gamma_{i(\ell)}^N(f) \cdot n_{i(\ell)}, \quad (7)$$

ove

$$m_{is}^f = \frac{\bar{M}_{is}^f}{\sum_i \sum_s \bar{M}_{is}^f} \quad \text{è il peso relativo del fattore } \bar{M}_{is}^f, \quad m_{is}^f \in \bar{M} = [0, 1];$$

$$h_{is} = \frac{H_{is}}{\sum_i \sum_s H_{is}} \quad \text{è il peso relativo del fattore } H_{is}, \quad h_{is} \in \bar{H} = [0, 1];$$

$a_i^f = \frac{A_i^f}{\sum_i A_i^f}$ è il peso relativo del fattore A_i^f , $a_i^f \in A = [0, 1]$;

$n_{i(\ell)} = \frac{N_{i(\ell)}}{\sum_i N_{i(\ell)}}$ è il peso relativo del fattore $N_{i(\ell)}$, $n_{i(\ell)} \in N = [0, 1]$

con $\gamma_{is}^M(f), \gamma_{is}^H(f), \gamma_i^A(f), \gamma_{i(\ell)}^N(f)$ funzioni che assumono valori nell'intervallo $[0, 1]$ $\forall x \in X$, $0 \leq \gamma_x(f) \leq 1$, ove X è l'insieme dei fattori.

Queste funzioni dipenderanno dalle caratteristiche e dalle preferenze che contraddistinguono ciascuna famiglia f . I valori di tali funzioni non sono altro che le valutazioni che la famiglia di tipo f esprime nei confronti dei singoli fattori.

$\gamma_{is}^M(f), \gamma_{is}^H(f), \gamma_i^A(f), \gamma_{i(\ell)}^N(f)$ possono essere riscritti come segue:

$$\gamma_{is}^M(f) = \mu_f(m_{is}^f)$$

$$\gamma_{is}^H(f) = \eta_f(h_{is})$$

$$\gamma_i^A(f) = \gamma_f(a_i^f)$$

$$\gamma_{i(\ell)}^N(f) = \delta_f[\bar{n}_{i(\ell)}]$$

ove $\mu_f(m_{is}^f), \eta_f(h_{is}), \gamma_f(a_i^f), \delta_f[\bar{n}_{i(\ell)}]$ esprimono il grado di ap-

partenenza di ciascun fattore alle funzioni di utilità per una famiglia di tipo f (esprimono, cioè, in che misura ciascun fattore contribuisce a determinare il livello di utilità di ogni famiglia).

La (7) rappresenta la funzione di utilità reale per una famiglia di tipo f , che occupa un alloggio di tipo s nella zona i .

2.1.4. Stabilità strutturale

Due aspetti non facilmente intuibili, e talora inattesi, del comportamento di un modello complesso, quale quello descritto in -per via analitica- si individuerà una condizione di instabilità;

2.1.2., sono:

- a. l'inerzia, che si riconosce quando, per grandi variazioni di alcuni parametri del modello, i risultati cambiano inapprezzabilmente;
- b. l'instabilità, che si riconosce quando, per piccole variazioni di alcuni parametri del modello, i risultati cambiano apprezzabilmente.

Se l'inerzia e l'instabilità trovano riscontro nel sistema modellato, ciò costituisce un contributo importante alla validazione del modello.

Per quanto precede, quando si opera con modelli complessi, l'analisi della sensibilità dei risultati, rispetto ai parametri,

diventa oggetto di fondamentale importanza, anche se, sovente, il gran numero di parametri stessi rende assai arduo, quando non impossibile (dato l'onere computazionale), condurre un'analisi sistematica.

Diventa, quindi, necessario cercare di conoscere a priori, per qualche via, il sottoinsieme di parametri rispetto ai quali il comportamento del modello può essere più critico.

A questo scopo si introduce qui una versione assai semplificata del modello descritto in 2.1.2., studiando le cui proprietà -per via analitica- si individuerà una condizione di instabilità, con riferimento alla quale diviene interessante sondare -per via simulativa- il comportamento del modello originale.

La semplificazione ha due aspetti principali:

- a. si focalizza l'interesse sul cuore del modello, ossia sul sottomodello residenziale e, in particolare, sulla mobilità residenziale presa in considerazione nel sottomodello;
- b. si elimina la disaggregazione sia socioeconomica sia spaziale.

Con ciò, detti:

- x il numero di famiglie che, ad un certo tempo, cambiano alloggio;
- y il numero di alloggi che, a quel tempo, sono disponibili per le famiglie che cambiano residenza;

il modello originale si può ridurre al seguente modello dinamico:

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{c}), \quad (8)$$

ove :

$\underline{x} = (x, y)$ è il vettore di stato;

\underline{c} è il vettore dei parametri;

\underline{f} è la funzione che definisce il cambiamento di stato.

Si fa osservare che x rappresenta anche il livello della mobilità residenziale ad un certo tempo.

Si avanza ora l'ipotesi che il modello dinamico sia lineare; la (8) prende, allora, la forma:

$$\dot{\underline{x}} = \underline{A} \underline{x} + \underline{b}, \quad (9)$$

ove gli elementi della matrice \underline{A} e del vettore \underline{b} sono i sei elementi del vettore dei parametri \underline{c} prima considerato.

E' possibile precisare il segno di alcuni parametri, introducendo le seguenti ipotesi (che l'osservazione suggerisce debbano valere sia per il modello semplificato sia per il modello originale):

- a. al crescere di x , coeteris paribus, \dot{x} decresce ed \dot{y} cresce
(in altre parole, quanto maggiore è l'aliquota delle famiglie che cambiano residenza, tanto minore è il tasso di crescita di

tale aliquota e tanto maggiore è l'offerta di alloggi);

b. al crescere di y , coeteris paribus, \dot{y} decresce ed \dot{x} cresce (in altre parole, quanto maggiore è l'aliquota degli alloggi disponibili per le famiglie che cambiano residenza, tanto minore è il tasso di crescita di tale aliquota e tanto maggiore è la domanda di abitazioni).

Si ha così:

$$\dot{x} = ay - mx + g \quad (10)$$

$$\dot{y} = bx - ny + h,$$

ove a , b , m ed n sono positivi e g ed h possono essere valori sia positivi sia negativi.

Il processo dinamico descritto dalla (10) è conosciuto come processo di Richardson (Richardson, 1960).

Posto quanto precede, si cominci con il considerare l'esistenza di punti di equilibrio nel processo dinamico, ponendo $\dot{x} = 0$ e $\dot{y} = 0$.

Si ha:

$$\begin{aligned} ay - mx + g &= 0 \\ bx - ny + h &= 0, \end{aligned} \quad (11)$$

che sono le equazioni di due rette L_1 ed L_2 nello spazio delle

fasi (si veda fig. 4):

$$\begin{aligned} L_1 : y &= (m/a) x - (g/a) \\ L_2 : y &= (b/n) x + (h/n). \end{aligned} \quad (12)$$

Si ha un punto di equilibrio quando L_1 ed L_2 si intersecano nel primo quadrante (si veda fig.4A e fig.4B) [infatti, le coordinate (x_0, y_0) del punto di equilibrio soddisfano entrambe le (11) ed è, come deve essere, $x_0 > 0$ e $y_0 > 0$].

Notato che i coefficienti angolari di L_1 e L_2 sono entrambi positivi, se (Braun, 1975):

a. il coefficiente angolare di L_1 è maggiore di quello di L_2 :

$$(m/a) > (b/n) \text{ ossia } mn - ab > 0, \quad (13)$$

si ha una soluzione di equilibrio (stabile) quando:

$$ng + ah > 0 \quad \text{e} \quad bg + mh > 0; \quad (14)$$

b. il coefficiente angolare di L_1 è minore di quello di L_2 :

$$(m/a) < (b/n) \text{ ossia } mn - ab < 0, \quad (15)$$

si ha una soluzione di equilibrio (instabile) quando:

$$ng + ah < 0 \quad \text{e} \quad bg + mh < 0. \quad (16)$$

Si possono riconoscere quattro tipi di comportamenti, a seconda che il vettore dei parametri \underline{c} :

A. soddisfa la (13) e soddisfa la (14) (si veda fig.4A);

B. soddisfa la (13) e non soddisfa la (14) (si veda fig. 4B);

C. soddisfa la (15) e soddisfa la (16) (si veda fig. 4C);

D. soddisfa la (15) e non soddisfa la (16) (si veda fig. 4D).

Le frecce in fig. 4 indicano la direzione di cambiamento del sistema.

Si osserva che, per ogni caso eccetto il C, per ognuna delle aree del piano delle fasi delimitate da L_1 e L_2 , si ha un'unica direzione di cambiamento. Per il caso C, ciò non si dà e la linea tratteggiata (separatrice) divide due direzioni alternative di cambiamento.

Dalle direzioni di cambiamento si trae, ancora, che l'equilibrio nel caso A è stabile, nel caso C è instabile.

In particolare, se i valori dei parametri del sistema sono tali per cui si è:

- nel caso A, il livello della mobilità residenziale tende ad assestarsi su un valore di equilibrio;
- nel caso B, il livello della mobilità residenziale va sempre più diminuendo;
- nel caso D, il livello della mobilità residenziale va sempre più aumentando;
- nel caso C, il livello della mobilità residenziale, a seconda delle condizioni iniziali del sistema, o va sempre più aumentando o va sempre più diminuendo o, assai improbabilmente, si

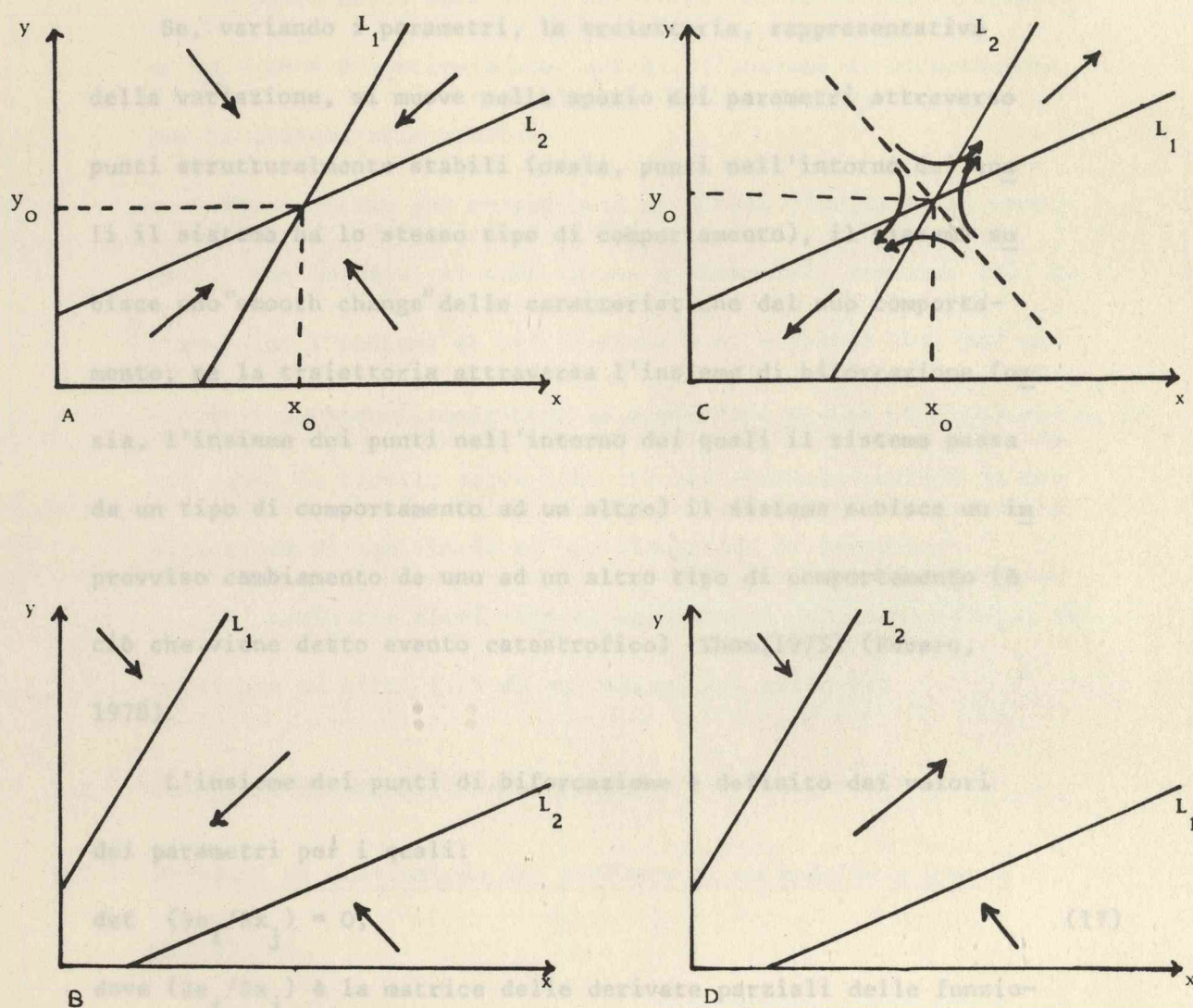


Figura 4 - Tipi di comportamento nel processo di Richardson

colloca su di un valore di equilibrio, da cui, anche per lievis
simi cambiamenti dei parametri, si allontana, ricadendo in una
delle due situazioni precedenti.

Se, variando i parametri, la traiettoria, rappresentativa
della variazione, si muove nello spazio dei parametri attraverso
punti strutturalmente stabili (ossia, punti nell'intorno dei qua
li il sistema ha lo stesso tipo di comportamento), il sistema su
bisce uno "smooth change" delle caratteristiche del suo comporta
mento; se la traiettoria attraversa l'insieme di biforcazione (os
sia, l'insieme dei punti nell'intorno dei quali il sistema passa
da un tipo di comportamento ad un altro) il sistema subisce un im
provviso cambiamento da uno ad un altro tipo di comportamento (è
ciò che viene detto evento catastrofico) (Thom, 1975) (Fararo,
1978).

L'insieme dei punti di biforcazione è definito dai valori
dei parametri per i quali:

$$\det (\partial e_i / \partial x_j) = 0, \quad (17)$$

dove $(\partial e_i / \partial x_j)$ è la matrice delle derivate parziali delle funzio
ni e_i , definite da:

$$e_i(x_1, x_2, \dots, c) = \frac{\partial x_i}{\partial t} = 0. \quad (18)$$

Per il processo di Richardson si ha:

$$\det \begin{pmatrix} -m & a \\ b & -n \end{pmatrix} = mn - ab = 0. \quad (19)$$

I punti nello spazio dei parametri che soddisfano l'equazione $mn - ab = 0$ costituiscono, quindi, l'insieme di biforcazione per il sistema allo studio.

Per mostrare una conseguenza di questo risultato, si consideri, come esempio, il caso in cui i parametri, variando (*), attraversino l'insieme di biforcazione e si supponga che, pur variando i parametri, sussistano le condizioni $ng + ah > 0$ e $gb + mb > 0$; in tal caso, il livello della mobilità residenziale passerà da una situazione di equilibrio ad una situazione di crescita.

E' ovvio che altri tipi di cambiamenti potranno prodursi in relazione ad altri tipi di variazioni dei parametri.

2.2. FORMULAZIONE OPERATIVA DEL MODELLO

2.2.1. Problemi di costruzione del software di un modello a grandi dimensioni

I problemi di costruzione del software del modello in oggetto

(*) Si ricorda che, nel modello descritto in 2.1.2., il sottomodello semplificato, che ora si sta analizzando, è legato ad altri sottomodelli, così che i valori dei parametri c sono definiti da questi altri sottomodelli e, quindi, possono variare nel tempo.

to verranno analizzati sotto gli aspetti di:

- A. qualità del software;
- B. gestione delle grandi dimensioni;
- C. gestione dell'input del modello;
- D. architettura complessiva del software per il modello.

Le considerazioni presentate si riferiscono alla particolare esperienza di costruzione del software relativo al presente modello. La loro applicabilità ad altri modelli simulativi, oltre che dalle analogie, presumibilmente verificabili, sulle caratteristiche dell'input (esogeno - dati iniziali ed eventuali dati di ciclo - ed endogeno) e sulla struttura interna (insieme di sub-modelli funzionali interconnessi richiamati ciclicamente con sequenza predeterminata), dipende soprattutto dalla soluzione adottata nel risolvere il problema della gestione delle strutture dati (array) "a grande dimensione" che caratterizzano appunto il modello; il che implica precise considerazioni sul linguaggio scelto e sulla relativa versione a disposizione, oltre alla ovvia constatazione della relatività delle "grandi" dimensioni all'estensione delle risorse software disponibili (essenzialmente, memoria centrale a disposizione dell'utente ed eventuali strumenti software che ne permettono l'ampliamento logico).

Più precisamente, il software per il suddetto modello è scritto (in massima parte) in FORTRAN IV ANSI (*), con l'utilizzo delle istruzioni relative ai files direct (***) e di altre caratteristiche non standard, in ambiente IBM.

A. Qualità del software

La qualità del software in generale è determinata da un insieme di caratteristiche, quali affidabilità, robustezza, correttezza ecc. (***), alcune relative all'intero sistema software, altre più strettamente connesse a singoli componenti di esso; inoltre, alcune, quali l'affidabilità, interessano direttamente l'attività di verifica della produzione ed installazione del software, mentre altre, quali la modificabilità, sono soprattutto determinate in fase di progetto. Le caratteristiche

-
- (*) American National Standard (1966) FORTRAN, ANSI, New York, X3.9..
 - (**) IBM System/360 and System/370, FORTRAN IV Language, order form GC28-6515.
 - (***) Per una rassegna sullo stato dell'arte relativamente alle ricerche ed agli strumenti sulla qualità del software, cfr.: Celentano, Ghezzy, Liguori (1980).

del modello di simulazione in oggetto hanno portato ad identificare le caratteristiche della qualità del software negli obiettivi di seguito trattati.

1. Correttezza

Alla accezione di essa, come grado di corrispondenza tra i risultati attesi, definiti in termini di specifiche di progetto, e quelli ottenuti, per una certa funzione (in senso lato) del software, occorre aggiungere che, dato il particolare carattere di modello di simulazione, alcuni risultati non sono prevedibili a priori con precisione. La correttezza viene allora ottenuta con la calibrazione del modello e, al limite, con variazioni (ovviamente, di limitata entità) nella sua stessa struttura. Detta caratteristica è quindi ancora più strettamente connessa con la modificabilità, che non in situazioni con specifiche di progetto sufficientemente definite.

Sono stati utilizzati, nelle fasi di design e codifica, approcci e strumenti logici complessivamente definibili come di programmazione strutturata (ad esempio: Myers, 1976; Wirth, 1972). Più precisamente, sono state effettuate:

- la decomposizione del modello del sistema complessivo e dei relativi sottosistemi in elementi funzionali cui corrisponde un ben determinato modulo di programma. In particolare, ad ogni equazione corrisponde una distinta porzione di codice;
- la eliminazione (o riduzione al minimo) dei collegamenti tra i moduli non attinenti ai legami funzionali;
- la stesura dei singoli moduli adottando una opportuna pseudo-codifica (Lanzarone, Maiocchi, Polillo, 1977), che utilizza costrutti di controllo strutturati, e codificando i singoli moduli in FORTRAN, facendo corrispondere ordinatamente gruppi di istruzioni equivalenti ai singoli costrutti strutturati.

2. Modificabilità

Sono stati considerati tre aspetti della modificabilità:

- variazioni dimensionali: le variazioni di grandezza, che esprimono valori dimensionali nel modello del sistema complessivo e dei relativi sottosistemi, devono avere effetto minimo o nullo sul codice; inoltre, l'utente deve poter intervenire su di essi fornendo i valori come da-

ti al programma stesso;

- variazioni strutturali e logiche del modello: quanto detto a proposito della correttezza implica che le variazioni nella logica e struttura del modello (nell'articolazione del sistema in sottosistemi, nelle equazioni) abbiano effetti soltanto sul codice corrispondente;
- variazioni sulla struttura dell'input e sulle modalità delle simulazioni: le informazioni sulla struttura dell'input (dimensioni, indici, caratteristiche temporali, allocazione) e sulla modalità degli esperimenti simulativi (intervallo temporale, inizializzazioni, input prescelti e loro caratteristiche) sono contenute in tabelle.

L'esecuzione delle equazioni del modello è quindi logicamente separata da quella di definizione degli esperimenti e di acquisizione degli input (esogeni) prescelti e quest'ultima lo è da quella di definizione generale degli input. Entro certi limiti, ciascuna delle tre può variare indipendentemente dalle altre.

3. Riproducibilità dei singoli esperimenti

Tutte le informazioni relative all'input, stato ed output del sistema simulato vengono memorizzate per ogni ciclo,

permettendo così di riprodurre, da qualunque ciclo, a condizioni identiche o mutate, ogni esperimento effettuato.

4. Portabilità

Per quanto non sia attualmente previsto un utilizzo del software su altri sistemi, un grado relativo di portabilità è stato ottenuto utilizzando generalmente caratteristiche standard nella codifica (FORTRAN IV ANSI), indicando con precisione le eccezioni ed isolando i gruppi di codice necessariamente dipendenti dal tipo di elaboratore utilizzato.

B. Gestione delle grandi dimensioni

Il modello in oggetto utilizza alcune grandezze con un notevole grado di analiticità; ne risultano quindi array con un numero molto elevato di elementi che, tradotti in codice, richiedono memoria centrale nell'ordine di parecchie decine di milioni di byte (*).

Almeno per il sistema operativo a disposizione (**), la

-
- (*) Il solo array DPOTO, distribuzione del numero delle famiglie (con capofamiglia occupato) secondo la zona di residenza, quella di lavoro, la tipologia familiare e la tipologia residenziale, ha 99x99x40x18 elementi, pari ad un numero di oltre 25 milioni di byte (nel caso gli elementi siano numeri reali di 4 byte).
 - (**) Il sistema operativo OS/VS su IBM non permette l'utilizzo di indirizzi virtuali maggiori di 16.777.216 byte (cfr.: IBM System/370, Principles of Operation, IBM, GA22-7000). Praticamente, nell'installazione utilizzata, non è normalmente disponibile una partizione superiore a 1536 Kbyte.

memoria centrale disponibile non è sufficiente alla esigenza indicata. Definiamo quindi (informalmente) come a grande dimensione un array tale da creare un modulo eseguibile incompatibile con la memoria centrale (virtuale) a disposizione.

1. Possibili soluzioni

In generale, il problema di esecuzione di un processo in uno spazio fisico S_f di lunghezza maggiore dello spazio S_i degli indirizzi (Bovet, 1977, p. 133 e segg.) può essere risolto secondo tre tecniche principali:

- ricoprimento manuale, ove i trasferimenti tra memoria centrale e secondaria sono gestiti dal programmatore;
- ricoprimento automatico, ove il programmatore indica la struttura dei ricoprimenti (overlay) e questa viene gestita durante l'esecuzione del lavoro (run-time) da appositi programmi di sistema;
- memoria virtuale: il programmatore codifica come se lo spazio fisico fosse compatibile con quello degli indirizzi; il sistema operativo gestisce automaticamente gli indirizzamenti e i trasferimenti tra memoria centrale e secondaria.

Scartata a priori l'ipotesi di utilizzare tecniche di ricoprimento manuale, che avrebbe comportato un notevole grado

di rigidità del codice alle variazioni dimensionali e strutturali del modello, oltre a complicare notevolmente il codice stesso, la scelta resta tra il ricoprimento automatico ed una qualche forma di virtualizzazione degli array a grande dimensione (considerando che il codice oggetto, aree dati escluse, è in sé ampiamente compatibile con la memoria centrale disponibile).

Il primo tipo di soluzione è stato utilizzato nel Detroit Prototype del modello di simulazione urbana del National Bureau of Economic Research (USA) (Ingram, Kain, Ginn, 1972). Questo modello, che richiede un minor spazio per i dati di quello in oggetto (ad esempio, le zone sono 44 contro 99), utilizza una struttura di overlay di 11 segmenti oltre al main, per 26 subroutine, codificate in FORTRAN IV, e 25 labeled common. Ogni subroutine si riferisce ad una o più labeled common contenente i dati richiesti, ed ogni segmento contiene almeno un gruppo subroutine + labeled common. La gestione è predisposta dal linkage editor (*).

(*) Cfr.: IBM OS/VS, Linkage Editor and Loader, order form GC26-3813.

Teoricamente una soluzione simile era possibile anche per il modello presente; tuttavia, alcuni array (come il già citato DPOTO) avrebbero da soli occupato più di un'area labeled common, riproducendo, sebbene in scala minore, i problemi del ricoprimento manuale. E' stata quindi messa a punto una particolare tecnica di indirizzamento che svincola il programmatore da problemi di indirizzamento e ricoprimento per gli array a grande dimensione.

2. La tecnica scelta

La memoria centrale richiesta dal programma, che è codificato prescindendo da concreti limiti di memoria, è simulata da un file FORTRAN ad accesso diretto (memoria esterna). Gli array a grande dimensione sono allocati in gruppi consecutivi di record di detta memoria esterna. Nella memoria centrale del calcolatore, più precisamente in una particolare labeled common area (memoria interna), risiedono soltanto alcuni segmenti (pagine) di alcuni array a grande dimensione, cioè di quelli utilizzati in un certo momento in una parte di programma (blocco). Se l'elemento richiesto di un determinato array è presente nella relativa pagina, esso viene messo a disposizione; altrimenti, detta pagina

viene scritta nelle opportune posizioni in memoria ester
na e la nuova pagina, contenente l'elemento richiesto,
viene caricata in memoria interna. Queste due operazio-
ni costituiscono un ciclo di paginazione.

In ciascun blocco, l'ampiezza della pagina è, in pri-
ma approssimazione, inversamente proporzionale al numero
degli array presenti in memoria interna. In realtà, poten-
do coesistere in uno stesso blocco differenti tipi di array
(reali, interi, logici, complessi, di diversa lunghezza),
le ampiezze delle pagine varieranno da tipo a tipo e da
blocco a blocco. Le funzioni di indirizzamento e paginazio-
ne indicate sono svolte da un insieme di sottoprogrammi,
presenti sotto forma di moduli oggetto in apposita libreria.

Per poter utilizzare detti sottoprogrammi, il program-
ma FORTRAN, codificato secondo quanto detto sopra, necessi-
ta di alcune trasformazioni. Queste sono effettuate aggiun-
gendo ad esso particolari istruzioni non-FORTRAN (che defi-
niscono gli array a grande dimensione, l'ampiezza delle me-
morie esterna ed interna, e delimitano i blocchi) e sotto-
ponendo il testo sorgente così ottenuto ad un apposito pro-

gramma, detto pre-compilatore, che produce come output un programma FORTRAN in grado di utilizzare la tecnica suddet-
ta.

Le limitazioni all'ampiezza della memoria interna sono ovviamente determinate dalla memoria centrale a disposizione, mentre nei riguardi della memoria esterna il programma precompilato è nella stessa situazione di un normale programma FORTRAN che richiede l'utilizzo di un file direct.

D'altro lato, ciascun array può avere, teoricamente, un numero di elementi dell'ordine del massimo intero positivo contenuto in una parola della memoria ($\approx 2^{31}$, nel nostro caso).

3. Il precompilatore

E' un traduttore, scritto a sua volta in FORTRAN IV ANSI, che effettua il processo suddetto sulla base di opportuni statement, aventi sintassi analoga a particolari istruzioni FORTRAN, i quali permettono di effettuare, oltre alle operazioni necessarie per l'utilizzo dell'algoritmo di indirizzamento, operazioni accessorie di input-output e di misurazione delle prestazioni del programma precompilato.

4. Limiti ed aspetti operativi della tecnica scelta

Pur essendo attualmente l'utilizzo del precompilatore in una fase sperimentale, sono apparsi critici, nel determinare le prestazioni del programma precompilato, i punti seguenti:

- numero di accessi agli array: deve essere minimizzato, compatibilmente con la logica del programma, attraverso l'utilizzo di opportune variabili transitorie, per evitare accessi non necessari alle routine di indirizzamento;
- sequenza di accessi agli array; per evitare fenomeni di trashing (cicli di paginazione continui), occorrono sequenze di accesso consecutive rispetto a quella di memorizzazione dei singoli elementi, che, coerentemente con la convenzione FORTRAN, è quella lessicale inversa;
- struttura dei blocchi (numero e tipo di array ivi presenti): deve essere tale da minimizzare il numero di blocchi nel programma (per evitare eccessive operazioni di allocazione e deallocazione).

C. Gestione dell'input del modello

Considerata la struttura della memoria esterna e l'esigenza, prima indicata, di utilizzare, negli array multidimen-

sionali, un ordine di variazione degli indici tale da minimizzare l'attività di paginazione, è stato predisposto, per l'esecuzione delle simulazioni, un sottosistema ad hoc di gestione dell'input.

Considerando che i dati grezzi originari possono trovarsi in supporti con caratteristiche di codifica e struttura record diversi per ciascun gruppo di dati o variabili, e tenendo presenti gli obiettivi di flessibilità indicati a proposito della qualità del software, la gestione dell'input viene effettuata attraverso due fasi distinte.

1. Normalizzazione dell'input

La memoria esterna, utilizzata dal programma (che traduce il modello di simulazione) opportunamente precompilato, ha un determinato modo di codifica (unformatted), una determinata lunghezza di record-blocco (stabilita in precompilazione secondo criteri di ottimizzazione dell'attività di paginazione), e gli array multidimensionali sono registrati secondo un determinato ordine di variazione degli indici (stabilito in fase di stesura delle equazioni del modello).

I dati grezzi, che, come si è detto, possono avere carat

teristiche di registrazione diverse, vengono quindi sottoposti alle due seguenti operazioni:

- trasposizione generalizzata (o riordino) dell'ordine di variazione degli indici attuale a quello desiderato;
- normalizzazione delle registrazioni (su disco, con lunghezza record desiderata, unformatted).

Sulla base di indicazioni (specifiche di input), relative al nome, tipo di variabile, dimensioni, indici (attuali e desiderati), formato di input, riferimenti temporali, viene ottenuto un file (input) che è in corrispondenza con la parte della memoria esterna che accoglierà i dati di inizializzazione e gli altri input esogeni di ciclo.

Accanto a questo file ne viene costruito un altro (catalogo dell'input), che contiene, per ogni variabile, oltre alle informazioni date dalle specifiche di input, la posizione su input.

Input e catalogo di input costituiscono l'input normalizzato al programma di simulazione.

2. Definizione dei singoli esperimenti

Data una certa versione del programma di simulazione, un esperimento, identificato da un nome, viene definito in

dicandone il periodo iniziale, quello finale e gli input esogeni nei relativi periodi.

Sono stati considerati quattro tipi di input:

- a- input di inizializzazione (input delle variabili di livello, assunti all'epoca iniziale della simulazione);
- b- costanti (valori costanti per tutti i periodi della simulazione);
- c- input periodici (input variabili da un periodo all'altro della simulazione secondo una legge esprimibile in termini di iterazione: inizio, fine ed incremento temporale positivo);
- d- input aperiodici (input variabili saltuariamente da un periodo all'altro della simulazione, secondo una legge esprimibile in termini di due o più iterazioni consecutive).

Da un punto di vista di calcolo gli input a e b differiscono solo per il fatto che i b per definizione non ammettono riassegnazioni successive. Verranno quindi ambedue considerati come input costanti (iniziali). Inoltre, il tipo c è assimilabile al tipo d qualora questo sia espresso da una o più iterazioni consecutive; verranno quindi, questi

ultimi considerati come input periodici.

Tutte le informazioni che definiscono un esperimento vengono poste in un file detto catalogo esperimenti: esse indicano, periodo per periodo, quali variabili trasferire da input alla memoria interna, e i relativi indirizzi (numero relativo di record).

3. Attività di input durante gli esperimenti

Ogni esperimento viene attivato indicandone il nome al programma di simulazione. Viene quindi acquisita, dal catalogo esperimenti, la relativa definizione e su questa base, in ciascun ciclo ove sono presenti, vengono effettuati i trasferimenti da input a memoria centrale.

Considerando che vi è corrispondenza tra le parti di input interessate e la memoria esterna, l'indicazione dei dati da trasferire è effettuata in termini di posizione in ciascuno dei due file, ottimizzando il trasferimento stesso.

D. Architettura complessiva del software per il modello

Considerando le indicazioni date in precedenza, è ora possibile delineare sia l'architettura complessiva del software sia le modalità di utilizzo.

In fig. 5 è schematizzato il software direttamente attinente l'esecuzione di esperimenti di simulazione. Sono stati indicati due moduli (input e simulazione) che compiono la funzione prima delineata. In fig. 6 è indicata la struttura del modulo di simulazione. Circa le modalità di costruzione e utilizzo dell'intero software sono stati effettuati i seguenti passi:

- costruzione e messa a punto del precompilatore;
- costruzione e messa a punto del modulo di input;
- precompilazione del sottomodello di simulazione;
- costruzione dei dati grezzi in input;
- normalizzazione degli input;
- definizione degli esperimenti;
- indicazione degli esperimenti.

Per quanto riguarda l'output dei singoli esperimenti, si possono distinguere tre tipi di dati:

- prospetti standard, generati al termine di ogni ciclo;
- run log, costituito da un file sequenziale in cui, per ogni esperimento, viene memorizzata la definizione, come da catalogo, e per ogni ciclo il valore iniziale di tutte le variabili, permettendo così la riproducibilità di un esperimento a partire da un ciclo scelto, e variandone eventualmente la

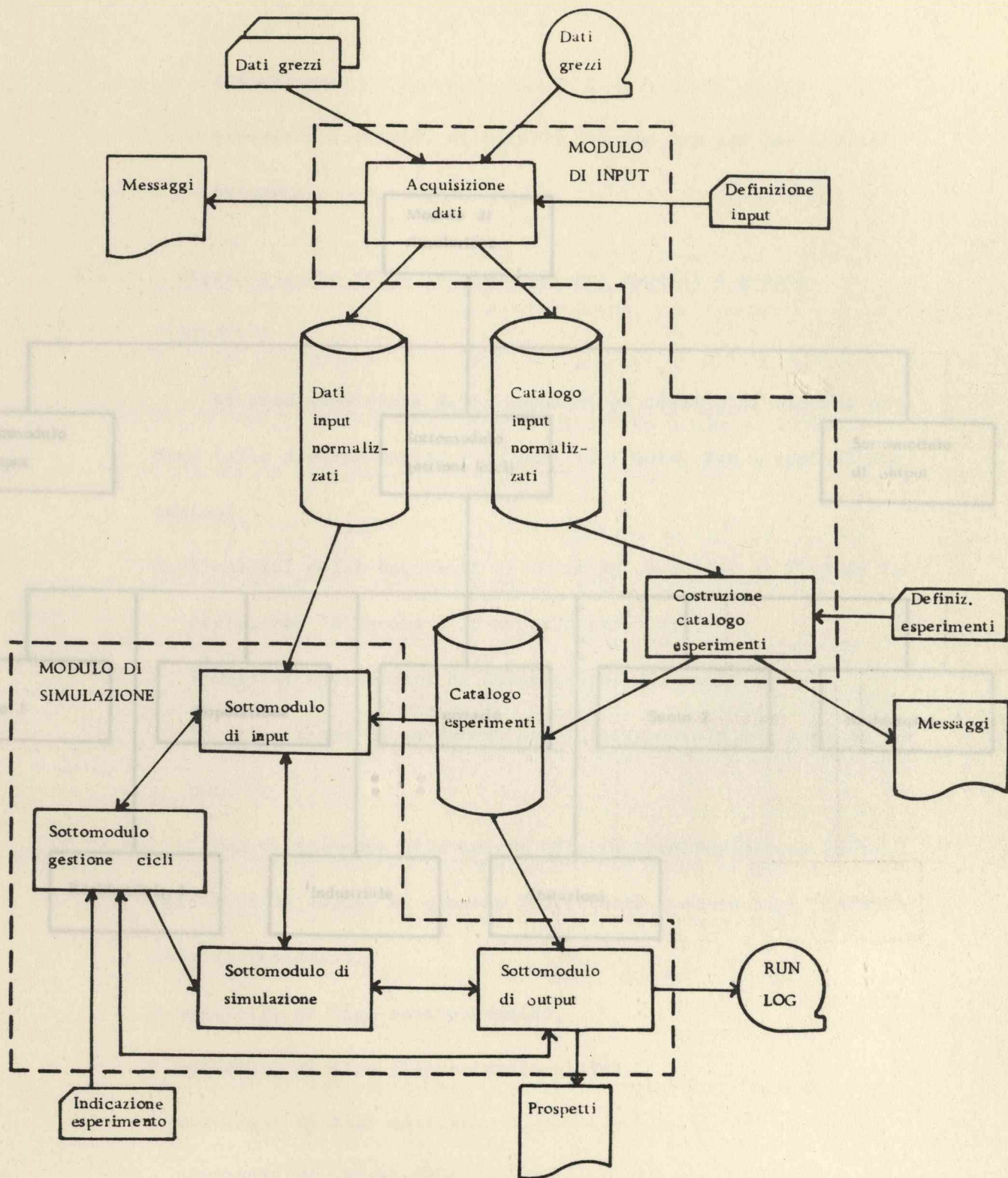


Figura 5 - Struttura del modulo di simulazione

Figura 5 - Architettura complessiva del software

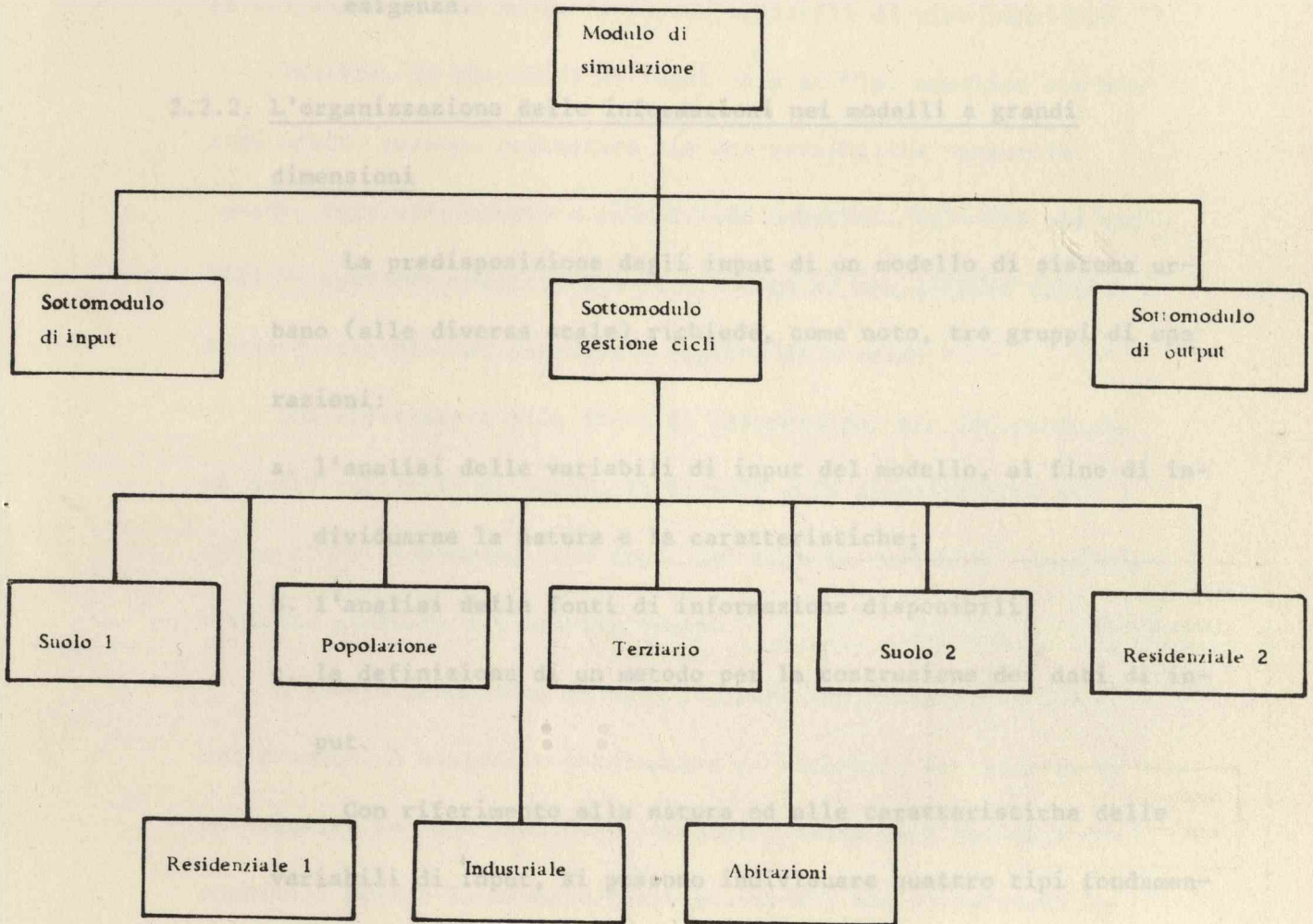


Figura 6 - Struttura del modulo di simulazione

definizione;
- prospetti speciali, ottenibili dal run log per particolari esigenze.

2.2.2. L'organizzazione delle informazioni nei modelli a grandi dimensioni

La predisposizione degli input di un modello di sistema urbano (alle diverse scale) richiede, come noto, tre gruppi di operazioni:

- a. l'analisi delle variabili di input del modello, al fine di individuare la natura e le caratteristiche;
- b. l'analisi delle fonti di informazione disponibili;
- c. la definizione di un metodo per la costruzione dei dati di input.

Con riferimento alla natura ed alle caratteristiche delle variabili di input, si possono individuare quattro tipi fondamentali di variabili:

- variabili di tipo socioeconomico;
- variabili di tipo fisico-territoriale;
- variabili di tipo edilizio;
- variabili di relazione.

Spesso, si è di fronte a variabili di input che, contempora-

neamente, possono avere le caratteristiche singolarmente sopra indicate e che, pertanto, diremo variabili di tipo misto (per cui le variabili di cui sopra le diremo variabili di tipo semplice).

Inoltre, le variabili di input (sia di tipo semplice sia di tipo misto) possono presentare sia una variabilità temporale -ossia, fare riferimento a più periodi temporali (*)- sia una variabilità spaziale -ossia, fare riferimento ad una qualche articolazione spaziale del territorio oggetto di studio-.

Con riferimento alle fonti di informazione per la costruzione delle variabili di input, il livello e la disponibilità di dette fonti dipendono, come noto, dal tipo di variabile -semplice o mista- presente nel modello stesso.

A questo proposito, si deve rilevare che, mentre, il più delle volte, è possibile predisporre le variabili di input di tipo semplice in modo pressoché immediato, sulla base dei dati disponibili presso le diverse fonti esistenti, non altrettanto agevole è procedere alla predisposizione delle variabili di input di tipo misto (e, in modo particolare, di quelle spazialmente e/o temporalmente disaggregate), per le quali le fonti disponi-

(*) A proposito della variabilità temporale degli input, si veda quanto già detto in 2.2.1..

bili si rivelano insufficienti e frammentarie, quando non addirittura inesistenti.

In generale, la predisposizione di variabili di input di tipo misto (e, in particolare, di quelle spazialmente disaggregate) richiede le seguenti operazioni:

- analisi e raccolta dei dati relativi alle variabili di tipo semplice costituenti la variabile mista;
- incrocio dei dati relativi alle variabili di tipo semplice.

In particolare, quest'ultima operazione è, fondamentalmente, un'operazione di assegnazione (che, naturalmente, deve risultare il meno arbitraria possibile) di un insieme di dati ad un altro insieme.

Per effettuare questa operazione, un metodo generale e relativamente semplice si fonda sui concetti di disaggregazione e di entropia.

Tramite la disaggregazione, si aumenta la numerosità delle zone alle quali le variabili di tipo semplice sono riferite (o, se si vuole, si riduce la dimensione delle zone territoriali originarie), ottenendo un'informazione spazialmente più dettagliata -cioè, più puntuale- delle variabili stesse e, dunque, un'informazione maggiore.

Tramite il metodo dell'entropia, tenendo conto di tutte le informazioni di cui si dispone in ordine alle variabili di tipo semplice, si ricerca l'assegnazione più probabile e meno arbitraria dell'insieme di dati -relativi ad una variabile di tipo semplice- ad un altro insieme di dati relativi a quella variabile.

In particolare, in questo lavoro, il metodo descritto è stato applicato per costruire la variabile di input X_{ifs} (numero di famiglie secondo la zona residenziale i , la classe familiare f e la tipologia residenziale s) (*).

Per ciascuna delle zone i (le quali sono più piccole di quelle originarie), si ricerca l'assegnazione più probabile di F_f (famiglie secondo la classe familiare f) a F_s (famiglie secondo la tipologia residenziale s), massimizzando l'entropia:

$$\text{Max } S = - \sum_s \sum_f \ln X_{of}!, \quad (20)$$

subordinatamente ai vincoli:

$$\sum_s X_{sf} = F_f \quad (21)$$

(*) Va precisato che la predisposizione di questo input ha costituito solo una fase della predisposizione delle variabili di input più complesse DPOTO (i, j, f, s) e DPOTD (i, g, s).

$$\sum_f X_{sf} = F_s \quad \text{che è una costante per ogni } s \text{ e per ogni } f, \text{ l'equazione (24) può essere riscritta, a meno di una costante di proporzione} \quad (22)$$

$$\sum_s \sum_f X_{sf} \cdot c_{sf} = C, \quad (23)$$

ove

X_{sf} è il numero di famiglie con tipologia residenziale s e classe familiare f ;

F_f è il numero di famiglie con tipologia familiare f ;

F_s è il numero di famiglie con tipologia residenziale s ;

c_{sf} è il costo di assegnazione residenziale di un alloggio di tipo s ad una famiglia di tipo f ;

C è il costo medio di assegnazione residenziale.

Risolvendo il sistema formato dalle equazioni (20),(21),(22)

e (23), si ottiene:

$$X_{sf} = A_f \cdot B_s \cdot F_f \cdot F_s \cdot \exp(-\gamma c_{sf}), \quad (24)$$

ove

$$A_f = \left[\sum_s B_s F_s \cdot \exp(-\gamma c_{sf}) \right]^{-1} \quad (25)$$

$$B_s = \left[\sum_f A_f F_f \cdot \exp(-\gamma c_{sf}) \right]^{-1}. \quad (26)$$

Ponendo:

$$Q_f = F_f \cdot A_f$$

$$R_s = F_s \cdot B_s$$

ed assumendo che c_{sf} sia costante per ogni s e per ogni f , l'equazione (24) può essere riscritta, a meno di una costante di proporzionalità, come segue:

$$X_{sf} = Q_f \cdot R_s$$

La (27) può essere interpretata come un'assegnazione biproporzionale ai termini Q_f e R_s .

Alonso W. (1984) Location and Planning, Mass.

Bayer D.P. (1977) Statistical operations, ISEI, Milano.

Braun M. (1975) Differential equations and their applications, Springer Verlag, New York.

Calabrese A., Ghezzi G., Liguori F. (1980) Qualità del software e metodologia di convalida, Rivista di Informatica, 3, 119-145.

Fararo T.J. (1978) An introduction to catastrophes, Behavioural Science, 23, 291-317.

Forrester J.W. (1969) Urban dynamics, M.I.T. Press, Cambridge, Mass.

Forrier G. (1978) Contribution à l'analyse spatiale de l'attraction imprécise, Document de travail, 30, I.N.E.S., Université de Dijon, Dijon.

Greenberg M.R. (1960) A model for the distribution of residential activity in urban areas, Journal of Regional Science, 2, 21-36.

BIBLIOGRAFIA

- Alonso W. (1964) Location and land use, Harvard University Press, Cambridge, Mass..
- Anas A. (1973) A dynamic disequilibrium model of residential location. Environment and Planning, 5, 633-647.
- Bertuglia C.S., Occelli S., Rabino G., Tadei R. (1980) A model of urban structure and development of Turin: theoretical aspects, Sistemi Urbani, 1.
- Bovet D.P. (1977) Sistemi operativi, ISEDI, Milano.
- Braun M. (1975) Differential equations and their applications, Springer Verlag, New York.
- Celentano A., Ghezzy G., Liguori F. (1980) Qualità del software e metodologia di convalida, Rivista di Informatica, 2, 119-143.
- Fararo T.J. (1978) An introduction to catastrophes, Behavioural Science, 23, 291-317.
- Forrester J.W. (1969) Urban dynamics, M I T Press, Cambridge, Mass..
- Fustier B. (1978) Contribution à l'analyse spatiale de l'attraction imprécise, Document de travail, 30, I.M.E., Université de Dijon, Dijon.
- Herbert J.T., Stevens H.H. (1960) A model for the distribution of residential activity in urban areas, Journal of Regional Science, 2, 21-36.

- Ingram G.K., Kain J.F., Ginn J.R. (1972) The Detroit Prototype of the NBER Urban Simulation Model, Columbia University Press, New York.
- Ires (1976) Linee di piano territoriale per il comprensorio di Torino, Guida Editori, Napoli.
- Lanzarone G.A., Maiocchi M., Polillo R. (1977) Introduzione alla programmazione strutturata, F. Angeli, Milano.
- Lowry I.S. (1964) A model of a metropolis, Rand Corporation, Santa Monica.
- Maciejowski J.M. (1978) The modelling of systems with small observation sets, Lecture Notes in Control and Information Sciences, 10, Springer Verlag.
- Myers G.J. (1976) Software Reliability, Wiley, New York.
- Ponsard C. (1978) On the imprecision of consumers' spatial preferences, Document de Travail, 26, I.M.E., Université de Dijon, Dijon.
- Richardson L.F. (1960) Arms and insecurity, Boxwood Press, Pittsburgh.
- Thom R. (1975) Structural stability and morphogenesis, Benjamin Press, Reading, Mass..
- Wilson A.G. (1970) Entropy in urban and regional modelling, Pion, London.
- Wilson A.G. (1974) Urban and regional models in geography and planning, Wiley, London.
- Wirth N. (1972) Systematisches programmieren, B.G. Teubner Verlag, Stuttgart.

ires

ISTITUTO RICERCHE ECONOMICO - SOCIALI DEL PIEMONTE
VIA BOGINO 21 10123 TORINO